TITLE OF THE INVENTION

IP Address Management

FIELD OF THE INVENTION

[0001] The present invention relates to an address management in networks in which addresses are allocated dynamically from a limited address pool. In particular, the invention relates to managing addresses in an access network assigning addresses to users of an IP network.

BACKGROUND OF THE INVENTION

[0002] IP (Internet Protocol) addresses, such as IPv4 (IP version 4) addresses are a limited resource that has to be recycled: when a user loses a connection, the address will be assigned to a new user. In case the old user has disconnected brutally, i.e. a connection is lost without having a chance to inform correspondent nodes about the connection loss, the address of the old user may have been reassigned to a new user before all of the old user's sessions have expired. As a result, the new user may receive packets that belong to the old user of the IPv4 address. For example, such connection losses are caused by the old user moving out of a coverage area or running out of battery.

[0003] Although the new user has not asked for these packets, he has to pay for receiving them anyway. This "overbilling" attack which happens e.g. in GPRS (General Packet Radio Service) and 3G (Third Generation) networks is a security incident.

[0004] Typically a stateful FW (FireWall) is used to stop incoming unsolicited packets. However, in the case of brutal disconnect the FW states ("pinholes") are not reset, because the FW has not been notified about the loss of the connection. Both TCP (Transport Control Protocol) and UDP (User Datagram Protocol) packets get through, if they match an old pinhole.

[0005] The time to wait until an address can be reassigned, i.e. a "cooling" time, varies, because there are no standard values for either TCP retry limit or UDP soft state lifetime. Typically TCP gives up only after several minutes of trying (depending on implementation), and UDP soft states live at least one minute (depending on FW settings). Sessions can also end if a NAT (Network Address (and Port) Translator) along the path resets an address-port binding, because it has not been used for some predefined time. Details about NAT can be found in P. Srisuresh et al.: "Traditional IP Network Address Translator (Traditional NAT)", Network Working Group, RFC 3022, January 2001. All of the above times can be different, and can be configured at different sites.

[0006] For example, an address cooling mechanism using IPv6 is described in applicant's WO 01/93540. However, previous suggestions of address cooling mechanisms have used only an estimate of the longest possible cooling period before an address can be reactivated, and used that to estimate the size of the required cooling queue. Such size may be uncomfortably large for IPv4.

[0007] Another suggestion is that the FW should read ICMP (Internet Control Message Protocol) error notifications transmitted to the senders, and close remaining pinholes to released addresses as described in applicant's US 60/479509. More details

about ICMP are described by J. Postel: "Internet Control Message Protocol", Network Working Group, RFC 792, September 1981.

[0008] Alternatively it has been suggested to use two FWs, one at Gn side of a GGSN (GPRS Gateway Serving Node) detecting PDP (Packet Data Protocol) context terminations, and another on the Gi side stopping packets to addresses that the first FW reports as unused. This solution is very expensive, because it requires installing FWs in many places. Also it is limited to GPRS and 3G networks.

[0009] Moreover, the FW suggestions may not help if a free address pool is small, and a new user happens to activate the same service that the old user had running. In this case the new user opens a pinhole that again lets the old user's session through.

SUMMARY OF THE INVENTION

[0010] It is an object of the invention to improve reassigning IP addresses to users in order to avoid extra packets for the users resulting e.g. in the overbilling attack.

[0011] According to one embodiment of the invention, a network device for managing addresses to be assigned to users of an IP network is disclosed. The network device includes at least one queue for holding released addresses and the network device is configured to detect that a packet has been addressed to a released address held in the at least one queue and to return the held address to which the packet has been addressed to an end of the at least one queue.

[0012] According to another embodiment of the invention, a method of managing addresses to be assigned to users of an IP network is disclosed. The method includes the steps of detecting that a packet has been addressed to a released address held in a

queue holding released addresses and returning the held address, to which the packet has been addressed, to an end of the queue.

[0013] According to another embodiment of the invention, a network device for forwarding IP data packets is disclosed. The network device is configured to receive a packet addressed to an unused address and to send an error notification to a network node for managing addresses, the error notification indicating the unused address.

[0014] According to another embodiment of the invention, a method of forwarding IP data packets is disclosed, where the method includes the steps of receiving a packet addressed to an unused address and sending an error notification to a network node for managing addresses, the error notification indicating the unused address.

[0015] According to a further embodiment of the invention, a system for managing addresses to be assigned to users of an IP network is disclosed. The system includes a first network node for managing addresses, where the first network node includes at least one queue for holding released addresses. The first network node is configured to detect that a packet has been addressed to a released address held in the at least one queue and to return the held address to which the packet has been addressed to an end of the at least one queue. Additionally, the system includes a second network node for forwarding IP data packets and the second network node is configured to receive a packet addressed to an unused address and to send an error notification to the first network node, the error notification indicating the unused address.

[0016] The invention may also be provided by a computer program product comprising software code portions to be executed by a computer, microprocessor or the like.

[0017] According to the invention, the size of a cooling queue will depend on averages, not maximum, because a cooling period of each address will adapt to stack implementations of correspondent nodes of an old user. Troublesome addresses connected to long-living sessions may be sent back to the end of the cooling queue several times. With the invention the number of cooling addresses can be reduced significantly, because a free address pool needs only a few "cold" addresses that rise quickly to the front of the queue.

[0018] Furthermore, the solution according to the invention is independent of FW and NAT locations and timers.

[0019] Moreover, the cooling method of the invention can be extended to DHCP-(Dynamic Host Configuration Protocol) assigned addresses. Therefore the invention works with any access network.

[0020] In the following the present invention will be described by way of preferred embodiments thereof taking into account the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] Fig. 1 shows an address management according to the invention in case it is detected that a packet has been sent to a released address.

[0022] Fig. 2 shows an address management according to the invention in case it is detected that an address has been released.

[0023] Fig. 3 shows an address management according to the invention in case an address out of released addresses is to be reassigned.

[0024] Fig. 4 shows a schematic block diagram illustrating an address management entity according to an embodiment of the invention.

[0025] Fig. 5 shows a schematic block diagram illustrating an address management entity according to another embodiment of the invention.

DESCRIPTION OF THE INVENTION

[0026] Figs. 1, 2 and 3 show an address management scheme according to the invention. An address management entity shown in these figures is responsible for assigning addresses to IP network users and comprises a cooling queue 10 for holding released addresses such as released IPv4 addresses. An address release may result from a connection loss which may be due to a brutal disconnect of a user.

[0027] The upper part of Fig. 1 shows the cooling queue 10 holding released addresses A1 to An. In case the address management entity detects that a packet has been addressed to a released address held in the cooling queue 10, such as to the released address A4, the released address A4 is shifted to the end of the cooling queue 10. The result of this cooling queue update is shown in the lower part of Fig. 1. As shown in the lower part of Fig. 1, the released addresses A5 to An have moved forward by one place to the front of the cooling queue 10 and the released address A4 is placed at the end of the cooling queue 10.

[0028] The upper part of Fig. 2 shows like the upper part of Fig. 1 the cooling queue 10 holding the released addresses A1 to An. In case the address management entity detects that an address of a user has been released, such as an address An+1, the released address An+1 is added to the end of the cooling queue. The result of this cooling queue update is shown in the lower part of Fig. 2.

[0029] The upper part of Fig. 3 shows like the upper part of Figs. 1 and 2 the cooling queue 10 holding the released addresses A1 to An. In case an address out of the

released addresses held in the cooling queue 10 is to be reassigned, e.g. because a free address pool is empty, the address at the beginning of the cooling queue is taken out as address to be reassigned. According to Fig. 3, since the address at the beginning of the cooling queue is the address A1, the address A1 is taken from the cooling queue 10 for reassigning. The result of this reassignment is shown in the lower part of Fig. 3.

[0030] As described above, the address management entity or Address Manager (AM) that is responsible for assigning addresses lets a released address "cool" until no packets to an old user having had the released address before are received. The cooling mechanism is adaptive: it uses the least-recently-used cooling queue 10. Released addresses go to the end of the cooling queue 10. In case e.g. a free address pool is empty, the first address in the cooling queue 10 (A1 according to Fig. 3) gets assigned to a next new user.

[0031] The idea is that each time an address already in the cooling queue 10 receives a packet, the address is returned to the end of the cooling queue 10. In addition, an error message (e.g. ICMP "not reachable") may be returned to a sender or source of the packet, e.g. a correspondent node of the old user, to inform it that sessions to the address are broken.

[0032] In case the address management entity is placed in a data path, like a GGSN, it can handle the queue internally. In other words, the address management entity may detect that a packet has been addressed to a released address held in the cooling queue 10 by receiving a packet addressed to the released address. Furthermore, the address management entity may detect that an address of a user has been released by detecting a loss of a connection which releases its address.

[0033] Fig. 4 shows an embodiment of the invention in which the address management entity or address manager is in the data path, i.e., the address manager and an AR (Access Router) are integrated like in a GGSN that manages its own address pool. In Fig. 4, the integrated address manager and access router block 30 is located in an access network, e.g. an IP based access network, providing access to an IP network for a user. The block 30 comprises the cooling queue 10.

[0034] In case the functions of the address management entity and those of the access router are combined as shown in Fig. 4, the loss of a connection releases its address, which is added to the end of the cooling queue 10 as shown in Fig 2. A next released address goes behind it, etc. But each time a packet arrives to a cooling address held in the cooling queue 10, the address is again returned to the end as shown in Fig. 1, and an ICMP error notification may be sent to the source of the packet by the integrated block 30.

[0035] Returning a released address repeats until all sessions that are bound to the released address have expired. After that the address can advance to the first position in the queue, and can then be assigned to a new user.

[0036] In case the address management entity and the data path are separated like e.g. in DHCP (Dynamic Host Configuration Protocol), an access router along the data path of the old user notifies the address management entity about lost connections, and also sends copies of error messages to the address management entity, the error messages notifying packets sent to unused addresses. The cooling queue is in the address management part. For details about DHCP it is referred to S. Alexander, R. Droms: "DHCP Options and BOOTP Vendor Extensions", Network Working Group, RFC 2132, March 1997.

[0037] In other words, in case the address management entity is not located in the data path of the old user, the address management entity may detect that a packet has been addressed to a released address held in the cooling queue 10 by receiving an error notification indicating an unused address. Furthermore, the address management entity may detect that an address of a user has been released by receiving a notification thereon.

[0038] In case the access router in the data path receives a packet addressed to an unused address it sends an error notification to the address management entity, the error notification indicating the unused address. In addition, the access router may also send an error notification to a source of the packet. Moreover, in case the access router detects a loss of a connection which releases its address, it sends a notification about the released and now unused address to the address management entity.

[0039] Fig. 5 shows a separated access router 42 and address management entity or address manager 41. The address manager 41 comprises the cooling queue 10. The access router 42 is located in the data path of the old user.

[0040] When the access router 42 loses a connection, it notifies the address manager 41 about the address that can be released. The address manager 41 puts the address in the cooling queue 10 as described above in connection with Fig. 2. If the access router 42 later receives a packet to an unused, i.e. released address, it sends an error notification to the address manager 41, which pushes the address to the end of the cooling queue 10 as shown in Fig. 1. The access router 42 may also send an ICMP error as it is done in the integrated case above.

[0041] The message from the access router 42 to the address manager 41 to release an address can be RADIUS (Remote Authentication Dial-In User Service 'Accounting

Stop'. The access router – address manager error notification protocol can be ICMP, but also a proprietary protocol is possible. Details about RADIUS can be found in C. Rigney et al.: "Remote Authentication Dial In User Service (RADIUS)", Network Working Group, RFC 2865, June 2000, and C. Rigney: "RADIUS Accounting", Network Working Group, RFC 2866, June 2000.

[0042] The invention has been explained using a single LRU (Least Recently Used) cooling queue 10 per address range. However, it is possible to classify a released address into a group out of at least two address groups, each address group having its own cooling queue holding released addresses, and to add the released address to the end of the queue of the classified group, the queues being given a priority order for re-assigning the released addresses held in the queues.

[0043] In other words, it is possible to classify the addresses as being "troubled", or connected to misbehaving nodes, e.g. those that ignore ICMP errors. For example, an address may be classified as being troubled in case transmission attempts for this address continue in spite of an ICMP error report, or the number of packets sent to this address exceeds a configurable limit. This classification may be carried out by the address management entity such as the integrated address manager and access router 30 or the address manager 41. Troubled addresses may be sent to a special queue by the address management entity which special queue is used for reassigning only when the normal queue (i.e. the cooling queue 10) is empty. Addresses in the queues given a higher priority than the special queue for "troubled" addresses will pass the troubled addresses. This way it is possible to avoid a situation where a non-cooled address gets reassigned, because it has not been accessed for some time, and during that time it has reached the front of the cooling queue 10.

[0044] It is to be understood that the above description is illustrative of the invention and is not to be construed as limiting the invention. Various modifications and applications may occur to those skilled in the art without departing from the true spirit and scope of the invention as defined by the appended claims.